

OHD General Programming Standards and Guidelines Peer Review Checklist

Reviewer's Name:		Peer Review Date:	
Project Name:		Project ID: Enter if applicable	
Developer's Name:		Project Lead:	
Review Files & Source code			
Code Approved			

This checklist is to be used to assess source code during a peer review. Items which represent the code being reviewed should be checked off.

Refer to the *OHD General Programming Standards and Guidelines* document for more complete descriptions and examples of the items listed below.

1. Internal Documentation

- _____ Comment block exists at the beginning of the source file containing at least the following information: filename, original author's name, file creation date, development group, names of routines contained in the file (if applicable), and a brief statement of the purpose of the software in the file.

- _____ Each subroutine or function in the file is preceded by a comment block which provides the following information: routine name, original author's name, routine's creation date, purpose of the routine, a list of the calling arguments (their types and what they do), a list of required files and/or database tables, the routine's return value, error codes and exceptions processed by the routine, and a modification history indicating when and by whom changes were made.

2. Programming Standards

- _____ Consistent indentation of at least 3 spaces is used to distinguish conditional or control blocks of code. TABS NOT USED FOR INDENTATION.
- _____ Inline comments are frequently used and adequately describe the code.
- _____ Structured programming techniques are adhered to.
- _____ Subroutines, functions, and methods are reasonably sized.
- _____ The routines in each source file shall have a common purpose or have interrelated functionality. Methods in a class support its functionality.

- _____ The name of the file, script or class represents its function.
- _____ Function and method names contain a verb, that is, they indicate an action.
- _____ Meaningful variable names are used.
- _____ Variables are initialized prior to use.
- _____ Braces are used consistently
- _____ There are no compiler warnings

3. Programming Guidelines

- _____ Source file line lengths are 80 characters or less.
- _____ Spacing is used correctly to enhance the source code's readability.
- _____ When continuing lines of code on new lines, they are broken after a comma or an operator. Higher level breaks are used instead of lower level breaks.
- _____ Wrapped lines of code are aligned with the beginning of the expression at the same level on the previous line.
- _____ Multiple line variable declarations are preceded by a type.
- _____ Program statements are limited to one per line.
- _____ Nested program statements are avoided.
- _____ Parentheses are used to remove operator precedence ambiguity and to improve code readability.
- _____ Inline comments constitute approximately 20% of the total lines of code in the program, excluding the file and routine documentation blocks.
- _____ The software reflects a balance of coding for efficiency and coding for readability.
- _____ Meaningful error messages are used.
- _____ System calls which acquire system resources are tested for error returns.
- _____ Routines and methods are contain no more than 200 executable lines.
- _____ The number of routines in a source file is kept to a minimum for procedural languages.

Reviewer's Comments: